Project structure

This page provides additional information about how this website is build and what each file does. You do not have to understand everything mentioned here yet, we will introduce the important concepts during the exercises as well. However, it is important to give you some context early on.

This website is build using SvelteKit, Svelte, and Bootstrap. We provide a short description of these frameworks, but also check out their documentation if something is not clear. SvelteKit is a modern website framework based upon Svelte. Essentially, you can think of Svelte as a tool for building an interactive web-page and SvelteKit as a tool for building an entire website out of those pages. Bootstrap is a collection of styling and javascript functionality for common website elements that make it easy to get a clean look quickly. If you see us using CSS classes that are not defined in the Svelte component, then they are probably from bootstrap. You can find what they do in their documentation.

The structure

The project contains quite a few configuration files in the root folder. You should not touch these files, but we will briefly explain what each file is for:

- LICENSE:
  The open source license for this project.
- README.md
  A markdown file describing the project and giving insturctions on how to use it.
- package.json
  Configures npm. Specifies the dependencies and tells npm what to do when we want to build the website.
- package-lock.json
  This file is automatically generated by npm and keeps track of all dependencies that are installed.
- .gitignore
  Configures which files git should not include in the repository.
- svelte.config.js
  Configures SvelteKit to run on specific hosting providers.
- jsconfig.json
  Tells SvelteKit where to look for our source files.
- eslintrc.cjs
  Configures the eslintrc linter, so it can give us warnings about our code.
- .prettierrc

  Configures prettier so it can automatically format our code.
- .npmrc
  Configures npm to use strict package versions for the dependencies.

In addition, two folders will be generated when you are working on the project. `node_modules` contains the javascript dependencies and `.svelte-kit` contains internal files of SvelteKit. If you ever run into an issue where SvelteKit is giving errors about files you have moved, you can solve that by removing the `.svelte-kit` folder and running `$: npm run build` .

Finally, the two folders that matter: `static` and `src` . `static` contains the files that should be accesible on the website directly, such as the favicon that is shown in the browser's tab. Use this folder if you want to be able to acces a file (f.i. `my_file.json`) within a web-page. Put the file in the static folder and load it from the URL `/my_file.json` .

The `src` folder contains all the files that make up the website.

- `src/lib`
  This folder contains all javascript functions and Svelte components that are intended to be re-used across the site. You can import files in this folder using the `$lib` short-hand notation. For example, the `CodeInline` component is located at `src/lib/ui/CodeInline.svelte` and can be imported using `import CC from '$lib/ui/CodeInline.svelte';`

- `src/routes`
  This folder contains the pages of the website. Basically, each `.svelte` file in `src/routes/` is a page on the website. The location of the the file within relative to the folder specifies the URL of the page. For instance, `src/routes/instructions/initial_setup.svelte` is accesible at https://datavis-exercises.vercel.app/instructions/intitial_setup. Files called `index.svelte` are special, they are accesible at the URL of the folder they are located in. For instance, `src/routes/week_0/index.svelte` has URL https://datavis-exercises.vercel.app/week_0. The `src/__layout.svelte` is also special, it specifies the parts of the website that are the same for all files. Finally, sometimes you may want to write components specifically for one web-page, and they cannot be re-used by other pages. In that case, you can prefix their name with an underscore to tell SvelteKit those files are not entire pages it can display directly.

  SvelteKit also supports endpoints, which are javascript files located under `src/routes` . These files specify functions that run on the server and return data. They can be used to access databases without leaking credentials to the user. You will not need them during this course.

Santiago Ormando (2158932) GitLab MIT License